Privacy Protection: *p*-Sensitive *k*-Anonymity Property

Traian Marius Truta, Bindu Vinay
Department of Computer Science, Northern Kentucky University
{trutat1, vinayb}@nku.edu

Abstract

In this paper, we introduce a new privacy protection property called p-sensitive k-anonymity. The existing k-anonymity property protects against identity disclosure, but it fails to protect against attribute disclosure. The new introduced privacy model avoids this shortcoming. Two necessary conditions to achieve p-sensitive k-anonymity property are presented, and used in developing algorithms to create masked microdata with p-sensitive k-anonymity property using generalization and suppression.

1. Introduction

The privacy of individuals is a challenging task in a digitized world. The amount of individual information collected by various data holders is continually increasing. To protect this large amount of personal data against intruders (individuals who want to use confidential information for malicious purposes) has become an increasingly difficult task.

The benefits that are drawn from the collected individual data are far too important for the society, and the trend of collecting individual data will never slow down. Many times the data collectors are trusted parties, and an individual will share his/her confidential information without any hesitation. The best example is in a healthcare organization. A physician must have complete access to a patient's medical history for the patient's benefit. The healthcare organization can use statistical analysis or data mining techniques to extract valuable information about its patients for research purposes. Many times the research is done by different organizations, and the control over the individual data is hard to enforce. It is possible that the individual information will be used in a wrong way. For instance, imagine that a pharmaceutical company links a group of individuals with their diagnostics. A targeted marketing program can be used on the individuals, and their privacy will be violated.

Legislators from many countries have tried to regulate the use and the disclosure of confidential information. Usually, privacy regulations forbid the release of attributes that clearly identify individuals, such as *Name* and *Social Security Number*. Even after their removal, these data sources may be matched with other public databases on attributes such as *Zip Code*, *Sex*, *Race* and *Birth Date*, to re-identify individuals who were supposed to remain anonymous [22]. Joining attacks are made easier by the availability of other, complementary, databases over the Internet.

In the U.S., for example, privacy regulations promulgated by the Department of Health and Human Services as part of the *Health Insurance Portability and Accountability Act (HIPAA)* protect the confidentiality of electronic healthcare information [8]. Similar privacy regulation exists in other domains. For instance, *Gramm-Leach-Bliley Financial Modernization Act*, enacted in 1999, requires financial institutions to disclose their privacy policies and allows consumers to choose the level of sharing of their personal information with third parties [7]. Other countries have promulgated similar privacy regulations. For example, the *Canadian Standard Association's Model Code for the Protection of Personal Information* [18] and the *Australian Privacy Amendment Act* [3] contain similar privacy regulations. Various privacy regulations analyzed from a database perspective are presented in [2].

Several techniques to avoid the disclosure of confidential information are presented in the literature [1, 25]. In this paper we focus on *k*-anonymity approach presented by Sweeney and Samarati [22, 19]. First, we describe *k*-anonymity privacy protection model. We show that *k*-anonymity protects against identity disclosure [11], but it fails to protect against attribute disclosure [11]. Second, we introduce a new privacy protection model called *p*-sensitive *k*-anonymity that extends the existing model and protects against both identity and attribute disclosure. We analyze several necessary conditions to achieve *p*-sensitive *k*-anonymity property, and we propose a method to create datasets with this property using generalization and suppression [22, 19]. In the end of the paper we illustrate on a publicly available dataset

[16] that *k*-anonymity property fails to protect all confidential information, and, therefore, *p*-sensitive *k*-anonymity is useful in real datasets.

2. Motivation for a new anonymity property

While the attributes that directly identify individuals such as Name and SSN are removed from the published microdata, other attributes, such as *ZipCode* or *Age*, which could lead to the possible identification of individuals, are usually released to the researchers. Unfortunately, an intruder may use record linkage techniques [26] between these attributes and external available information to glean the identity of individuals from the released microdata. To avoid this possibility of disclosure, the data owner must release a modified version of the microdata that protects the identity of individuals and, simultaneously, is useful to researchers. We will refer to the released microdata as *masked microdata* and to the initial dataset as *initial microdata*. There are several methods (also called disclosure control or masking methods) presented in research papers (*sampling* [20], *global and local recoding* [14, 19, 23], *suppression and local suppression* [19, 13], *microaggregation* [5], *simulation* [1], *adding noise* [9], *randomization or perturbation methods* [15, 10, 6], *data swapping* [4, 17], *substitution* [21] etc.) to modify the initial microdata in order to achieve individual privacy and preserve data usefulness. While applying these methods, the data owner should decide where to draw the line between modifying the initial microdata too much (the useful information may be lost) or too little (some individuals may be still at high risk of disclosure). One solution proposed in the literature for the highly sensitive microdata to protect the identity of individuals, is to enforce a property that must hold for the masked microdata called *k*-anonymity [22, 19].

First, the data owner determines the set of attributes that do not directly identify an individual, but used in conjunction with other data sources may lead to disclosure of an individual. These attributes are called *quasi-identifiers* [22] or *key attributes* [24]. To simplify our discussion we use the following classification that includes all possible attributes from any microdata:

- $I_1, I_2,..., I_m$ are identifier attributes such as *Name* and *SSN* that can be used to identify a record. These attributes are present only in the initial microdata because they express information which can lead to a specific entity.
- $K_1, K_2,..., K_p$ are key attributes such as *Zip Code* and *Age* that may be known by an intruder. Key attributes are present in the masked microdata as well as in the initial microdata.
- $S_1, S_2,..., S_q$ are confidential attributes such as *Principal Diagnosis* and *Annual Income* that are assumed to be unknown to an intruder. Confidential attributes are present in the masked microdata as well as in the initial microdata.

To protect the data, the identifiers attributes are completely removed, and the key attributes are “masked”, using disclosure control methods, in order to avoid the possibility of disclosure. We assume that the values for the confidential attributes are not available from any external source. This assumption guarantees that an intruder can not use the confidential attributes values to increase his/her chances of disclosure, and, therefore, their “masking” is unnecessary.

Definition 1 (*k-anonymity property***)**: The *k-anonymity property* for a masked microdata (*MM*) is satisfied if every combination of key attribute values in *MM* occurs *k* or more times.

Based on this definition, in a masked microdata that satisfy *k*-anonymity property, the probability to identify correctly an individual is at most 1/*k*. By increasing *k* the level of protection increases, along with the changes to the initial microdata. In Table 1, we show an example of masked microdata where 2-anonymity is satisfied.

Table 1. *Patient* masked microdata satisfying 2-anonymity

Age	ZipCode	Sex	Illness
50	43102	M	Colon Cancer
30	43102	F	Breast Cancer
30	43102	F	HIV
20	43102	M	Diabetes
20	43102	M	Diabetes
50	43102	M	Heart Disease

In this example, the set of key attributes is composed of *Age*, *ZipCode*, and *Sex*. A simple SQL statement helps us check whether a relation adheres to *k*-anonymity:

SELECT COUNT(*) **FROM** *Patient* **GROUP BY** *Sex*, *ZipCode*, *Age***.**

If the results include groups with count less than k, the relation *Patient* does not have *k*-anonymity property with respect to *KA* = {*Age*, *ZipCode*, *Sex*}.

Using this example, we illustrate why *k*-anonymity does not provide the amount of confidentiality required for every individual. To justify this affirmation, we distinguish between two possible types of disclosure, namely**,** identity disclosure and attribute disclosure. *Identity disclosure* refers to identification of an entity (person, institution) and *attribute disclosure* occurs when the intruder finds out something new about the target entity [11]. Identity disclosure does not automatically imply attribute disclosure. It may happen that the intruder does not find anything new when he identifies an entity.

Also we can have attribute disclosure without identity disclosure. We illustrate these two concepts by using the *Patient* masked microdata from Table 1. There is no identity disclosure in this microdata, its construction guarantees that for every existing combination of values for *Age*, *ZipCode*, and *Sex*, there are at least two tuples that share the same combination of key attribute values, therefore, the masked microdata is protected against identity disclosure. We assume that the external information from Table 2 is available to a presumptive intruder, and the intruder also knows that in the masked microdata (see Table 1) the *Age* attribute was generalized to multiples of 10.

Table 2. External information for *Patient* example

Name	Age	Sex	ZipCode
Sam	29	M	43102
Gloria	38	F	43102
Adam	51	M	43102
Eric	29	M	43102
Tanisha	34	F	43102
Don	51	M	43102

The intruder does not know if Sam or Erich maps to the first or the second tuple from the *Patient* masked microdata with *Age*, *ZipCode*, and *Sex* combination of (20, 43102, M), but he does know that both of the tuples have Diabetes as the illness, and therefore both Sam and Erich have Diabetes. This example shows that *k*-anonymity privacy protection does not consider the attribute disclosure, and fails to protect individuals' privacy in certain situations. To avoid this problem, we generalize *k*-anonymity to a new privacy protection model called *p*-sensitive *k*-anonymity.

Definition 2 (*p-sensitive k-anonymity property*): The masked microdata (*MM*) satisfies *p-sensitive k-anonymity property* if it satisfies *k*-anonymity, and for each group of tuples with the identical combination of key attribute values that exists in *MM*, the number of distinct values for each confidential attribute occurs at least *p* times within the same group.

To illustrate this property, we consider the masked microdata from Table 3 that satisfies 3-anonymity property with respect to *Age*, *ZipCode* and *Sex*. To find the value of *p*, we analyze each group with identical values for all key attributes. The first group (the first three tuples) has two different illnesses, and only one income, therefore the value of *p* is 1. This masked microdata satisfies 1-sensitive 3-anonymity property. It is easy to notice that when *p* = 1 we will always have the attribute disclosure problem. If the first tuple would have a different value for income (such as 40,000) then both groups would have two different illnesses and two different incomes, and the value of *p* would be 2.

Table 3. Masked microdata example for *p*-sensitive *k*-anonymity property

Age	ZipCode	Sex	Illness	Income
20	43102	F	AIDS	50,000
20	43102	F	AIDS	50,000
20	43102	F	Diabetes	50,000
30	43102	M	Diabetes	30,000
30	43102	M	Diabetes	40,000
30	43102	M	Heart Disease	30,000
30	43102	M	Heart Disease	40,000

From the definition of *p*-sensitive *k*-anonymity property we notice that *p* is always less than or equal to *k*. From the above examples, we draw the following two conclusions:

- To avoid the possibility of identity disclosure, a given masked microdata must have *k*-anonymity property with *k* greater than or equal to 2.
- To avoid the possibility of attribute disclosure, a given masked microdata must have *p*-sensitive *k*-anonymity property with *p* greater than or equal to 2.

From these two properties, we can draw the wrong conclusion that 2-sensitivity 2-anonymity property will suffice to protect any masked microdata against disclosure. Unfortunately, in this case, an intruder may "guess" the identity or attribute value of some individuals with a probability of ½. For many masked microdata such a high probability is unacceptable, and the values of *k* and/or *p* must be increased.

3. *P*-sensitive *k*-anonymity property

Several algorithms that automatically modify a given initial microdata into masked microdata that satisfies the *k*-anonymity property are presented in the literature [12, 19]. The disclosure control methods used most frequently in this process are *generalization* [19] (also known as global recoding [14]) and *suppression* [19, 13].

Generalization is used with categorical attributes such as *ZipCode* and *Sex*. The domain for an attribute that is generalized is extended to a *domain generalization hierarchy*, which includes all possible groups for that specific attribute. For the attribute *ZipCode*, the domain contains all existing zip codes, while the domain generalization hierarchy contains all prefixes (without repetition) for the existing values [19]. A domain generalization hierarchy is a total ordered relation between different domains that can be associated with an attribute. The values from different domains can be represented in a tree called *value generalization hierarchy*. We illustrate domain and value generalization hierarchy in Figure 1.

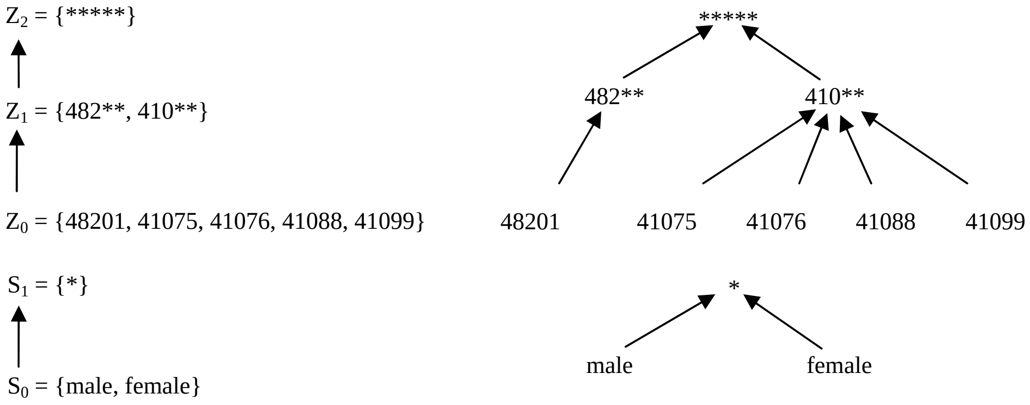

Figure 1. Examples of domain and value generalization hierarchies

To apply generalization, the data owner defines the domain and value generalization hierarchies for the attributes he wants to generalize. Usually, the data owner has several choices based on the properties of each attribute. For instance, the *ZipCode* attribute can have a different generalization hierarchy with six domains in which only one digit is removed at a time. The choice of the domain generalization hierarchy (the value generalization hierarchy is generated based on the chosen domain generalization hierarchy) is an important factor in the success of the masking process.

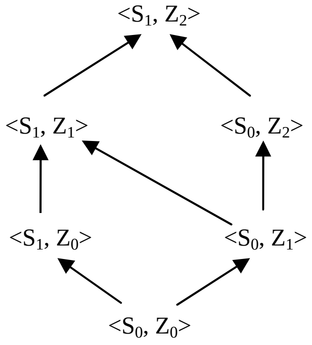

Figure 2. Generalization lattice for *ZipCode* and *Sex* attribute

When two or more attributes are generalized the data owner can create a *generalization lattice* to visualize all possible combinations of generalized domains (see Figure 2). Generalization lattices are introduced by Samarati [19]. The minimum path from the minimal element in the generalization lattice *GL* to a node *X* is labeled *height(X, GL)*. For instance in figure 2, $height(<S_0, Z_0>, GL) = 0$, $height(<S_1, Z_0>, GL) = 1$, $height(<S_0, Z_1>, GL) = 1$, $height(<S_1, Z_1>, GL) = 2$, $height(<S_0, Z_2>, GL) = 2$, $height(<S_1, Z_2>, GL) = 3$. The maximum height, represents the height of the entire lattice, and we label it *height(GL)*.

This generalization method (also called full domain generalization [12] or global recoding [McGulkin et al. 1990]) maps the entire domain of a key attribute in initial microdata to a more general domain from its domain generalization hierarchy.

Using only generalization, any initial microdata can be transformed to a masked microdata that satisfies *k*-anonymity. Practical experiments have shown that the generalization required to achieve *k*-anonymity considerably reduces the usefulness of the data, and the resulting masked microdata frequently will be useless. To avoid this shortcoming, one more disclosure control method called suppression is used.

After generalization is performed, we can identify the number of tuples that have a frequency of key attribute values less than *k*. If this number is below a defined threshold we apply suppression, and these tuples will be removed from the resulting masked microdata.

Using generalization and suppression we can obtain several distinct masked microdata sets that satisfy *k*-anonymity property. It is easy to prove that if *k*-anonymity is achieved for a node *X* in the generalization lattice, *k*-anonymity is satisfied for every node *Y* that is a generalization of the note *X* (*Y* is on the path from *X* to the upper level of the lattice) [19]. From the construction of the lattice we know that on every path we lose information when we move up in the lattice. Therefore, the data owner is interested in finding the node or nodes that are closer to the bottom of the lattice. A node *X* that satisfies *k*-anonymity when no other node *Y* satisfies *k*-anonymity such that *X* is on the path from *Y* to upper level of the lattice (*X* different of *Y*) represents a *k-minimal generalization* [19]. The data owner wants to find one or all *k*-minimal generalization.

For an initial microdata we have different *k*-minimal generalization based on the threshold selected for suppression. In Figure 3, we illustrate the *Sex* and *ZipCode* values from an initial microdata, and we compute (value in parentheses) in the associated generalization lattice, how many tuples do not satisfy 3-anonymity for every node. For every initial microdata and every generalization lattice, the number of tuples not satisfying *k*-anonymity decreases when the amount of generalization increases. Therefore, on every path this number increases as we traverse from the upper level node to the bottom.

Sex	ZipCode
M	41076
F	41099
M	41099
M	41076
F	43102
M	43102
M	43102
F	43103
M	48202
M	48201

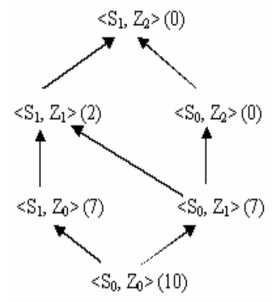

Figure 3. Example for minimal generalization with suppression threshold (TS)

The Table 4 shows which node corresponds to 3-minimal generalization for different values for the threshold (TS). We notice that the 3-minimal generalization is not unique for all threshold values.

Table 4. 3-minimal generalizations and TS values for suppression

TS	0, 1	2, 3, 4, 5, 6	7, 8, 9	10
Node	<S_0, Z_2>	<S_0, Z_2> and <S_1, Z_1>	<S_1, Z_0> and <S_0, Z_1>	<S_0, Z_0>

There are several algorithms available to determine a *k*-minimal generalization with a suppression threshold [19, 12, 22]. We improve these algorithms to find a minimal generalization (labeled *p-k-minimal generalization*) that satisfies *p*-sensitive *k*-anonymity property.

Definition 3 (*P-k-minimal generalization*)**:** A node *X* that satisfies *p*-sensitive *k*-anonymity when no other node *Y* satisfies *p*-sensitive *k*-anonymity such that *X* is on the path from *Y* to upper level of the lattice (*X* different of *Y*) represents a *p-k-minimal generalization*

The *p*-sensitive *k*-anonymity property can not always be satisfied. We consider the number of distinct values for each confidential attribute S_j (j = 1, …, q) as s_j. In this case *p* must always be less than or equal to $\min_{j=1,q}(s_j)$. For instance, if we consider *Sex* as a confidential attribute, that the number of distinct values for *Sex* is only 2, and therefore the maximum possible value for *p* is 2. The following condition must hold in order to have *p*-sensitive *k*-anonymity property for a masked microdata.

Condition 1 (*First necessary condition for a MM to have p-sensitive k-anonymity property*)**:** The minimum number of distinct values for confidential attributes must be greater than or equal to *p*.

It is easy to check if *p*-sensitive *k*-anonymity property can be obtained for a given *p*. For each confidential attribute the following SQL statement is executed for the initial microdata *IM* to find the number of distinct values (s_j): **SELECT** COUNT (*distinct* S_j) **FROM** *IM***.**

Next, the minimum value is determined and compared with *p*. If *p* is strictly greater than the selected value, the *p*-sensitive *k*-anonymity property cannot be satisfied.

We derive a second necessary condition for a MM to have the *p*-sensitive *k*-anonymity property. This condition establishes the maximum allowed number of combinations of key attribute values in the masked microdata that satisfy *p*-sensitive *k*-anonymity. To illustrate this condition, we consider a masked microdata *MM* with 1,000 tuples that has only one confidential attribute *S*. The attribute *S* has 5 distinct values which occur with the following frequencies: 900, 90, 5, 3, and 2. Also, *MM* has *k*-anonymity property for a fixed *k* greater than 3. Now, we would like to check whether *MM* has 3-sensitive *k*-anonymity property. To have this property each group determined by the combination of key attribute values must have three distinct values for the attribute S. We notice that if the number of such groups is 11 or more this property will never be true. We use the following definition for the *frequency set*.

Definition 4 (*Frequency set*)**:** Given a microdata *M* (initial or masked), and a set of attributes *SA* of *M*, the *frequency set* of *M* with respect to *SA* is a mapping from each unique combination of values of *SA* to the total number of tuples in *M* with these values of *SA* [12].

We use the following notations for a microdata *M*:

- *n* – the number of tuples in *M*;
- *q* – the number of confidential attributes in *M*;
- s_j – the number of distinct values for the attribute S_j; ($1 \le j \le q$)
- $f_i^{\,j}$ – the descending ordered frequency set for the confidential attribute S_j; ($1 \le j \le$ q and $1 \le i \le s_j$)
- $cf_i^{\,j}$ – the cumulative descending ordered frequency set for the confidential attribute S_j; ($1 \le j \le q$ and $1 \le i \le s_j$)
- $cf_i = \max_{j=1,q}(cf_i^{\,j})$ $(1 \le i \le \min_{j=1,q}(s_j)$)

To illustrate these notations we consider the following example.

Example 1: Consider a microdata *M* with two key attributes (K_1 and K_2) and three confidential attributes (S_1, S_2, and S_3). We assume the size of the microdata is 1,000. The Tables 5 and 6 summarize the frequency sets values introduced before.

We know from the first necessary condition that *p* must be less or equal to 5. We analyze for all possible values of *p* how many groups of distinct combinations of key attribute values are allowed for this masked microdata.

For $p = 2$ there are at most 300 groups allowed. The attribute S_3 has a value that repeats 700 times (cf_1), and one tuple with a different value must be included in every group.

When $p = 3$, the maximum allowed number of groups is 100, and when $p = 4$ the number of groups is at most 50. The justification is similar to the previous case.

A more interesting situation occurs for $p = 5$. The four most common values for S_3 occur 960 times, and to create groups of 5 distinct values we must include one of the remaining tuple in every group. The maximum number of groups seems to be 40 ($1000 – cf_4$). The problem is that we don't know if we already have 4 distinct values in every group. The most frequent three distinct values for S_3 represent 950 tuples, and we must include at least 2 of the remaining 50 tuples in every group. Therefore the maximum number of groups is only 25. We must repeat this reasoning for the most frequent 2 values, and for the most frequent value. Using the above justification, we have computed the minimum between the following values: $n - cf_{p-1}$, $\left\lfloor \frac{n - cf_{p-2}}{2} \right\rfloor$, .., $\left\lfloor \frac{n - cf_1}{p-1} \right\rfloor$; each of these representing an upper threshold for the maximum number of allowed groups. Based on this example we derive the following necessary condition.

Condition 2 (*Second necessary condition for a MM to have p-sensitive k-anonymity property***):** The maximum allowed number of combinations of key attribute values in the masked microdata is $\min\limits_{i=1,p-1} \left\lfloor \frac{n - cf_{p-i}}{i} \right\rfloor$.

Without using the two necessary conditions we can test for *p*-sensitive *k*-anonymity property using the following basic algorithm (see Algorithm 1). We can use the SQL statement: **SELECT** COUNT (*) **FROM** *MM* **GROUP BY** *KA* to determine if *k*-anonymity is satisfied for the masked microdata *MM*.

The Algorithm 1 is used in the process of searching for a masked microdata that satisfies *p*-sensitivity *k*-anonymity. Until such a masked microdata is found, there are many unsuccessful tries. We can improve this basic algorithm using the two necessary conditions (see Algorithm 2).

In multiple runs from different masked microdata sets created for the same initial microdata we can reuse the values *maxP* and *maxGroups* if the only disclosure control method applied was generalization of key attributes. Moreover, these two values can be computed from the initial microdata since all involved attributes are confidential, and the generalization method does not change these attributes. In case of generalization followed by suppression, it seems that the computation of *maxP* and *maxGroups* must be repeated for every masked microdata.

We prove the following two theorems that allow us to reuse the values of *maxP* and *maxGroups* for masked microdata where suppression was applied after the generalization method.

Theorem 1: Let *maxP* be the minimum number of distinct values for confidential attributes computed for the initial microdata *IM*, and $maxP_M$ the same value computed for the masked microdata *MM* derived trough generalization followed by suppression from *IM*. The inequality $maxP \geq maxP_M$ is always true.

Proof: Let s_j be the number of distinct values for the attribute S_j ($1 \leq j \leq q$) in *IM*, and s'_j be the number of distinct values for the attribute S_j ($1 \leq j \leq q$) in *MM*.

During the generalization, the values for confidential attributes do not change, and the existing changes are due only to suppression. The number of distinct values for any S_j cannot increase by eliminating tuples. Therefore, we have the following property:

$s_j \geq s'_j$, for all j, $1 \leq j \leq q$.

Let S_k be the confidential attribute with the smallest number of distinct values for *IM*. We have (using the above inequality):

$$maxP = \min_{j=1,q}(s_j) = s_k \geq s'_k \geq \min_{j=1,q}(s'_j) = maxP_M$$

q.e.d.

Table 5. Frequency set values

	s_j	f_1^j	f_2^j	f_3^j	f_4^j	f_5^j	f_6^j	f_7^j	f_8^j	f_9^j	f_{10}^j
$j = 1$	5	300	300	200	100	100	-	-	-	-	-
$j = 2$	6	500	300	100	40	35	25	-	-	-	-
$j = 3$	10	700	200	50	10	10	10	10	5	3	2

Table 6. Cumulative frequency set values

	s_j	cf_1^j	cf_2^j	cf_3^j	cf_4^j	cf_5^j	cf_6^j	cf_7^j	cf_8^j	cf_9^j	cf_{10}^j
$j = 1$	5	300	600	800	900	1000	-	-	-	-	-
$j = 2$	6	500	800	900	940	975	1000	-	-	-	-
$j = 3$	10	700	900	950	960	970	980	990	995	998	1000
cf_i	-	700	900	950	960	1000	-	-	-	-	-

Algorithm 1 (*Basic Algorithm to test the p-sensitive k-anonymity property for MM***):**

```
Input:  MM – a masked microdata
        p, k (p ≤ k) natural numbers greater than or equal to 2.

Output:   Condition is true (p-sensitive k-anonymity is satisfied)
          Condition is false (p-sensitive k-anonymity is not satisfied)

if MM has k-anonymity property then
{
      Condition = true;
      for each combination of key attribute values and each confidential attribute do
      {
            Let d be the number of distinct values for that confidential attribute.
            If d < p then
            {
               Condition = false;
               Break loop;
            }
       }
}
else Condition = false;
```

Algorithm 2 (*Improved Algorithm to test the p-sensitive k-anonymity property for MM***):**

```
Input:  MM – a masked microdata
        p, k (p ≤ k) natural numbers greater than or equal to 2.

Output:   Condition is true (p-sensitive k-anonymity is satisfied)
          Condition is false (p-sensitive k-anonymity is not satisfied)

Condition = true;

// first necessary condition
if Condition then
{
      Compute $s_j$ for all confidential attributes $S_j$ ($j$ = 1, ..., $q$)
      maxP = $\min_{j=1,q}(s_j)$

      if ( p > maxP ) Condition = false;
}

// second necessary condition
if Condition then
{
      Compute $f_i^j$, $cf_i^j$ for all confidential attributes $S_j$ (j = 1, ..., $q$)
      Compute $cf_i = \max_{j=1,q}(cf_i^j)$ for all $i$, 1 ≤ $i$ ≤ $\min_{j=1,q}(s_j)$

      maxGroups = $\min_{i=1,p-1}\left\lfloor \frac{n - cf_{p-i}}{i} \right\rfloor$

      Compute the number of combinations of key attribute values in MM.
          Let noGroups be this value.
      if (noGroups > maxGroups ) Condition = false;
}


if (Condition and MM does not have k-anonymity property) then Condition = false

// only the MM that pass the two conditions are checked in details.
for each combination of key attribute values and each confidential attribute do
{
       Let d be the number of distinct values for that confidential attribute.
       If d < p then
       {
         Condition = false;
         Break loop;
       }
}
```

Theorem 2: Let *maxGroups* be the maximum allowed number of combinations of key attribute values computed for the initial microdata *IM*, and $maxGroups_M$ the same value computed for a masked microdata *MM* derived trough generalization followed by suppression from *IM*. Then $maxGroups \geq maxGroups_M$

Proof: We use the notations *n'*, s'_j, f'^{j}_{i}, cf'^{j}_{i} and cf'_i for MM to distinguish them from *n*, s_j $f_i^{\,j}$, $cf_i^{\,j}$ and cf_i used for IM in this proof. We also define *ts* = *n* – *n'* as the suppression threshold. When suppression is applied to a microdata, the data owner usually determine the maximum allowed number of tuples to be removed. Therefore, without the loss of generality, we assume that *ts* is known from the start of the masking process.

From the definition of $cf_i^{\,j}$ and by removing at most *ts* tuples we have:

$0 \leq cf_i^{\,j} - cf'^{j}_{i} \leq ts$ for all *j*, $1 \leq j \leq q$ and for all *i*, $1 \leq i \leq \min_{j=1,q}(s_j)$ (since $cf'^{j}_{i} \leq \min_{j=1,q}(s_j)$, we initialize $cf'^{j}_{i} = n'$ for any *i* greater than cf'^{j}_{i})

Next, we use the second part of the above inequality:

$cf_i^{\,j} \leq cf'^{j}_{i} + ts$ for all *j*, $1 \leq j \leq q$ and for all *i*, $1 \leq i \leq \min_{j=1,q}(s'_j)\cdot$

For a fixed *i*, let S_k be the confidential attribute such that: $cf_i^{\,k} = \max_{j=1,q}(cf_i^{\,j})$. Then:

$$cf_i = \max_{j=1,q}(cf_i^{\,j}) = cf_i^{\,k} \leq cf'^{k}_{i} + ts \leq \max_{j=1,q}(cf'^{j}_{i}) + ts = cf'_i + ts.$$

From $cf_i \leq cf'_i + ts$ we get:

$$-cf_i \geq -cf'_i - ts\,, \qquad n - cf_i \geq n - ts - cf'_i\,,$$

$$n - cf_i \geq n' - cf'_i\,, \qquad \left\lfloor \frac{n - cf_{p-i}}{i} \right\rfloor \geq \left\lfloor \frac{n' - cf'_{p-i}}{i} \right\rfloor,$$

$$\min_{i=1,p-1} \left\lfloor \frac{n - cf_{p-i}}{i} \right\rfloor \geq \min_{i=1,p-1} \left\lfloor \frac{n' - cf'_{p-i}}{i} \right\rfloor,$$

$$maxGroups \geq maxGroups_M$$

q.e.d.

Algorithm 3 (*Algorithm for computing a p-k-minimal generalization*):

```
Input:  IM – an initial microdata
        GL – the generalization lattice
        p, k (p ≤ k) natural numbers greater or equal to 2.

Output: Condition is true (p-sensitive k-anonymity can be achieved)
        Condition is false (p-sensitive k-anonymity can not be achieved)
        X – the node in the lattice that represents the masked microdata with
           p-k-minimal generalization (only when Condition is true)

// first necessary condition can be checked from the beginning
Compute maxP using IM;
if ( p > maxP ) Condition = false;
else
{   Compute maxGroups for IM;  // second necessary condition preparation
    low = 0;
    high = height(GL);
    while (low < high)
    {
        try = (low + high) /2;
        Nodes = {Y | height(Y, GL) = try};
        reach_k = false;
        while (Nodes ≠ ∅) and (reach_k ≠ true)
        {
           Select and remove a node Z from Nodes.
           Compute the number of combinations of key attribute values for the MM that
             corresponds to Z using only generalizations. Let noGroupsZ be this value.
           if (noGroupsZ > maxGroups ) break; // second necessary condition

           if (p-sensitive k-anonimity property is satisfied
             for MM (including suppression) using Basic Algorithm)
           {
               reach_k = true; X = Z;
           }
          }
          if (reach_k==true) high = try;
          else low = try + 1;
     }
}
```

Based on the above theorems we can generalize any algorithm that finds a *k*-minimal generalization to an algorithm that finds a *p*-*k*-minimal generalization.

To illustrate our affirmation we consider the algorithm introduced by Samarati that uses binary search executed on the generalization lattice. For each node in the generalization lattice, we know the generalization that is applied to each key attribute, and we can apply our improved algorithm to test *p*-sensitive *k*-anonymity property. In the Algorithm 3, we underline our additions to the existing algorithm [19].

By introducing these two necessary conditions, we eliminate many masked microdata sets that do not satisfy *p*-sensitive *k*-anonymity property early. Also, these two necessary conditions can be used in correlation with other algorithms that computes masked microdata sets with *k*-anonymity property only [12].

4. Experimental results

In our experiments we used the *Adult* database from the UC Irvine Machine Learning Repository [16]. We considered *Age*, *MaritalStatus*, *Race*, and *Sex* as the set of key attributes, and *Pay*, *CapitalGain*, *CapitalLoss*, and *TaxPeriod* as the set of confidential attributes. We applied generalization for the key attributes using the generalization domains as described in the Table 7.

We use the following notations:

- A_i (i = 0, 1, 2, 3) for the domain generalization of *Age*;
- M_j (j =0, 1, 2) for *MaritalStatus*;
- R_k (k = 0, 1, 2, 3) for *Race*;
- S_p (p = 0, 1) for *Sex*.

We label a node in the generalization lattice (labeled GL_A) as $<A_i, M_j, R_k, S_p>$ where i = 0, 1, 2, 3; j =0, 1, 2; k = 0, 1, 2, 3; and p = 0, 1. The total number of nodes in the lattice is 4 x 3 x 4 x 2 = 96, and $height(GL_A)$ = 9.

We used two sample sets from the Adult database as our initial microdata sets, the first with size 400 and the second with size 4000. We applied Samarati binary search algorithm [19], and we found the node in the lattice that corresponds to the minimal *k*-generalization (the corresponding masked microdata has *k*-anonymity property). We choose *k* as either 2 or 3. Next, we analyze the values for confidential attributes within each group with the same key attribute values. In three out of four experiments we found several groups of attributes with the same value for a confidential attribute (which means that the masked microdata does not have 2-sensitive *k*-anonymity property), and therefore the attribute disclosure could take place. The results of our experiments are summarized in the Table 8.

This experiment shows the possibility of attribute disclosure when *k*-anonymity model is enforced, and the need for an enhanced model to avoid such a shortcoming. The proposed *p*-sensitive *k*-anonymity property guards against attribute disclosure. From this experiment, we found that when the value of k increases, the number of attributes disclosure decreases, which means, *p*-sensitivity *k*-anonymity property is satisfied for larger values of *p*. Although, this is true for many datasets, by choosing a larger *k* the attribute disclosure problem is not avoided.

5. Conclusions and future work

Our main contribution in this paper is the introduction of the *p*-sensitive *k*-anonymity property. This property is the first method that a data owner can use to protect the initial microdata against the attribute disclosure. Moreover, this property also includes the identity disclosure protection (*k*-anonymity.

Our experiments shows how *k*-anonymity fails to protect against attribute disclosure, and this motivated us to extend the *k*-anonymity model to include this enhanced level of protection.

Table 7. Adult database key attribute generalizations

Attribute	Distinct Values	First Generalization	Second Generalization	Third Generalization
Age	74	10-years ranges	<50 and >50 groups	One group
MaritalStatus	7	Single or Married	One group	-
Race	5	White, Black, or Other	White or Other	One group
Sex	2	One group	-	-

Table 8. Attribute disclosures for two masked microdata sets with k-anonymity property

Size and k-anonymity	Lattice Node	No of attribute disclosures
400 and 2-anonymity	$<A_1, M_1, R_1, S_1>$	6
400 and 3-anonymity	$<A_1, M_1, R_2, S_1>$	2
4000 and 2-anonymity	$<A_2, M_1, R_1, S_1>$	4
4000 and 3-anonymity	$<A_2, M_1, R_2, S_1>$	0

We found two important necessary conditions a masked microdata must satisfy in order to have *p*-sensitive *k*-anonymity property. Also, we proved two theorems which create the framework to use the necessary conditions efficiently in any algorithm that searches automatically for a masked microdata. We illustrated the inclusion of the two necessary conditions in the algorithm for computing a *p*-*k*-minimal generalization (see Algorithm 3).

In future work, we will create masked microdata that satisfy *p*-sensitive *k*-anonymity using the existing algorithms for *k*-anonymity with the addition of the two necessary conditions, and we will compare the running time of these modified algorithms against the existing algorithms that searches for *k*-anonymity only.

10. References

[1] N. R. Adam and J. C. Wortmann, “Security Control Methods for Statistical Databases: A Comparative Study”, *ACM Computing Surveys*, Vol. 21, No. 4, 1989.

[2] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, “Hippocratic Databases”, *Proc. of the 28th Int’l Conference on Very Large Databases*, Hong Kong, China, 2002.

[3] APA, “The Australian Privacy Amendment Act”, Available at *www.privacy.gov.au/publications/npps01.html*, 2000.

[4] T. Dalenius and S. P. Reiss, “Data-Swapping: A Technique for Disclosure Control”, *Journal of Statistical Planning and Inference*, Vol. 6, 1982, pp. 73-85.

[5] J. Domingo-Ferrer and J. Mateo-Sanz, “Practical Data-Oriented Microaggregation for Statistical Disclosure Control”, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 14, No. 1, 2002, pp. 189-201.

[6] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, “Privacy Preserving Mining of Association Rules”, *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 217-228.

[7] GLB, “Gramm-Leach-Bliley Financial Services Modernization Act”, Available at *www.banking.senate.gov/conf*, 1999.

[8] HIPAA, “Health Insurance Portability and Accountability Act”, Available at www.hhs.gov/ocr/hipaa, 2002.

[9] J.J. Kim, “A Method for Limiting Disclosure in Microdata Based on Random Noise and Transformation”, *American Statistical Association, Proceedings of the Section on Survey Research Methods*,1986, pp. 303-308.

[10] P. Kooiman, L. Willemborg, and J. Gouweleeuw, “PRAM: A Method for Disclosure Limitation for Microdata”, *Report*, Department of Statistical Methods, Statistical Netherlands, Voorburg, 1997.

[11] D. Lambert, “Measures of Disclosure Risk and Harm”, *Journal of Official Statistics*, Vol. 9, 1993, pp. 313-331.

[12] K. LeFevre, D. DeWitt, and R. Ramakrishnan, “Incognito: Efficient Full-Domain K-Anonymity”, *ACM SIGMOD International Conference on Management of Data*, 2005.

[13] R.J.A. Little, “Statistical Analysis of Masked Data”, *Journal of Official Statistics*, Vol. 9, 1993, pp. 407-426.

[14] R.H. McGuckin and S.V. Nguyen, “Public Use Microdata: Disclosure and Usefulness”, *Journal of Economic and Social Measurement*, Vol. 16, 1990, pp. 19 – 39.

[15] K. Muralidhar and R. Sarathy, “Security of Random Data Perturbation Methods”, *ACM Transactions on Database Systems*, Vol. 24, No. 4, 1999, pp. 487-493.

[16] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz, “UCI Repository of Machine Learning Databases, Available at *www.ics.uci.edu/~mlearn/MLRepository.html*, University of California, Irvine, 1998.

[17] S.P. Reiss, “Practical Data-Swapping: The First Steps”, *ACM Transactions on Database Systems*, Vol. 9, No. 1, 1984, pp. 20-37.

[18] M. Rotenberg, *The Privacy Low Sourcebook 2000: United States Law*, International Law, and Recent Developments, Electronic Privacy Information Center, 2000.

[19] P. Samarati, “Protecting Respondents Identities in Microdata Release”, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 13, No. 6, 2001, pp. 1010-1027

[20] C.J. Skinner, C. Marsh, S. Openshaw, and C. Wymer, “Disclosure Control for Census Microdata”, *Journal of Official Statistics*, 1994, pp. 31-51.

[21] A.C. Singh, F. Yu, and G.H. Dunteman, “MASSC: A New Data Mask for Limiting Statistical Information Loss and Disclosure”, *Joint ECE/EUROSTAT Work Session on Data Confidentiality*, Luxembourg, 2003.

[22] L. Sweeney, “Achieving *k*-Anonymity Privacy protection Using Generalization and Suppression”, *International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems*, Vol. 10, No. 5, 2002, pp. 571 – 588.

[23] A. Takemura, “Local Recoding by Maximum Weight Matching for Disclosure Control of Microdata Sets”, *ITME Discussion Paper*, No.11, 1999.

[24] T.M. Truta, F. Fotouhi, and D. Barth-Jones, “Disclosure Risk Measures for Microdata”, *Int’l Conference on Scientific and Statistical Database Management*, 2003, pp. 15-22.

[25] L. Willemborg, and T. Waal, *Elements of Statistical Disclosure Control*, Springer Verlag, 2001.

[26] W.E. Winkler, “Advanced Methods for Record Linkage”, *Proceedings of the Section on Survey Research Methods*, American Statistical Association, 1994, pp. 467-472.